RoboCup2005
Rescue Robot League Competition
Osaka, Japan
July 13 – 19, 2005
www.robocup2005.org

RoboCupRescue - Robot League Team
Autonomous relief, RFC Uppsala, Sweden

Torkel Danielsson, Markus Elving, Fredrik Hellman, Henrik Hofling,
Robert Koskinen, Mikael Laaksoharju, Kaveh Mehrabi, Carl Johan Otterheim,
Dan Pettersson, Per Svensson, Stefan Warnelid, Erik Winter, Mikael Carlsson, Jakob Carlström,
Per Halvarsson, Mattias Wiggberg

Uppsala University
Department of Information Technology
Box 337, SE-751 05 Uppsala, Sweden
jakob.carlstrom@it.uu.se

http://www.robocup.it.uu.se/itp2005/ringhorne

Abstract. The robots designed by Autonomous relief (RFC Uppsala) constitute a fully autonomous distributed robot system for solving rescue tasks. This year's robots are much improved, with a new powerful AI engine, more sensors and more accurate positioning routines. The new AI incorporates all sensor data available to all robots in a team into one accurate map. Dead reckoning pose estimate is now improved by SLAM. Our robots carry a multitude of sensors, including ultrasonic-, infrared-, CO_2- and pyro-electric sensors. The robots are also equipped with new cameras, using advanced stereo algorithms to extract data. All of these improvements are used when navigating through a disaster area while mapping it and quickly and reliably locating victims.

Introduction

Autonomous relief (RFC Uppsala) is a team from Uppsala University, consisting of last-year students from the MSc program in Information Technology together with senior staff from the university. RFC Uppsala has a long history in the RoboCup competitions. In 2004 we developed a fully autonomous rescue robot system to participate in the Rescue Robot League.

This year the autonomous robot system has been much improved, based on the experiences from last year. The rescue task consists of investigating a disaster area, constructing a map, and marking positions of any human victims on the map. This year's further development of the robot system was carefully planned. We wanted to improve the areas that proved most challenging last year, including navigation and victim detection in environments consisting of transparent materials. We made the decision to keep most of the robot chassis from last year as it worked satisfactory.

This years focus has been to improve the AI and the sensors. The AI is completely rebuilt; it explores faster, handles larger areas and constructs better maps than before. The new AI was designed using standard approaches, making the new system powerful, reusable and improvable.

All sensors used last year were reviewed and compared to alternative sensors, before the decision was made which to use. Many sensors are available on the robots and the AI makes use of the information from all of them.

Our own robot arena, built by an earlier student project in accordance with the NIST specifications, is a huge benefit in terms of measuring performance of the robots. We now have the possibility of improving the design under conditions replicating the competition.

To make the robot hardware and software more flexible, we now use the Player API [1]. This enables us to develop the hardware control and the AI at the same time by taking advantage of the many simulators and GUIs written for Player.

1. Team Members and Their Contributions

- Advisor — Jakob Carlström
- Advisor — Mikael Carlsson
- Advisor — Per Halvarsson
- Advisor — Mattias Wiggberg
- Project manager/ SW designer — Fredrik Hellman
- Technical coordinator — Torkel Danielsson
- AI designer/Image analysis — Markus Elving
- AI designer/Image analysis — Mikael Laaksoharju
- AI designer/Image analysis — Kaveh Mehrabi
- AI designer/Image analysis — Erik Winter
- Data communication/HW designer — Robert Koskinen
- Data communication/HW designer — Carl Johan Otterheim
- Data communication/HW designer — Dan Pettersson
- HW designer/OS — Per Svensson
- HW designer/OS — Henrik Hofling
- HW designer/OS — Stefan Warnelid

2. Operator Station Set-up and Break-Down

The whole team of robots can be controlled by a single operator working at a terminal acting either as a group leader giving general directions or having full control over each robot. The operator station consists of a terminal with the human/robot interaction software installed and a small portable printer, for printing the generated map. Since the robots are fully autonomous the operator station is not needed during missions, however it is possible to control the robots manually if desired.

Contrary to the RFC Uppsala 2004 Robot Rescue Team the robots can start from an arbitrary position and each robot can create a map based upon the starting position independently of where the other robot starts.

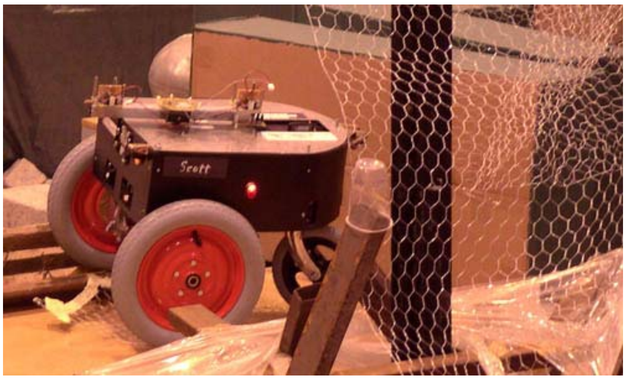

Fig 2.1. Autonomous robot during rescue mission.

3. Communications

The robots use 802.11a WLAN communication using an implementation of TCP which improves the performance of a wireless link, especially in congested network environments. The protocol works by adaptively setting a slow start threshold and a congestion window which takes into account the bandwidth used at the time congestion is experienced. Communication between the robots is done in infrastructure mode. Compared to last year this implementation has greatly improved the communication.

Rescue Robot League		
Autonomous Relief, RFC Uppsala (Sweden)		
Frequency	**Channel/Band**	**Power (mW)**
5.2 GHz - 802.11a	36/LOW OR 40/LOW OR 44/LOW OR 48/LOW	40

4. Control Method and Human-Robot Interface

The team of robots is designed to be fully autonomous. A human/robot interface is available when supervision is desirable although no human operator is needed during missions. Autonomous robot control is achieved using a three level data abstraction approach (see figure 4.1).

On the bottom level, the robot features a reactive obstacle-avoiding behavior. Raw sensor readings from close objects result in virtual forces on the robot, steering it free of obstacles and moving objects in its path.

On the intermediate level a map is constructed using all available sensor data. The top level is a mesh of nodes connected by paths where the nodes are strategically placed during the exploration and each path between two nodes has a cost associated with it. Navigation on the node level is obtained by searching the mesh using an A* based heuristic search. The autonomous exploration algorithm we use in this project is inspired by the frontier based exploration approach of Yamauchi [2].

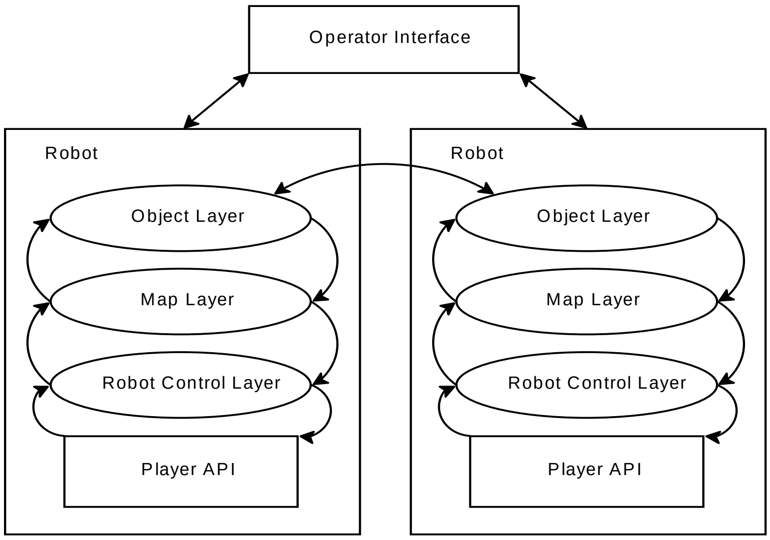

Fig. 4.1. System architecture.

A human/robot interface can be used to control the whole team of robots from one single terminal over the WLAN interface. Status information of all subsystems for each robot and advanced functions for motion control is available from the human/robot interface. The operator can choose to control the robots' exact paths, but can also let the robots calculate their own paths and search behavior if that is preferred. This allows the operator to act as a supervising team leader merely guiding the robots to particular sites of interest.

5. Map generation/printing

The map is generated autonomously, simultaneously with the self localization of the robots within it. We have chosen to represent the map as an evidence grid [3]. The map generating algorithm works in more than one stage and uses the probabilistic mathematical framework shared by most current robot mapping algorithms [4]. A robot uses new sensor data to adjust its pose before adding the information to the map.

In autonomous mode, the robots explore and map their surroundings incrementally. A team of autonomous robots exploring an area does so without designating one robot as master. The team of robots shares a common map if the relative positions of the robots are known.

Algorithms are also implemented to allow robots, exploring the same area, whose relative starting locations are unknown to fuse their maps. This fusing of maps is automated and will occur when two robots with unknown relative positions encounter each other. To effectively explore and map an unknown area, a team of robots will share information about the frontiers visible from their current positions. The task of exploring these frontiers are autonomously divided amongst the team members. In this way the exploration of an area will be more effective the more members an exploring team has.

The lack of a master/slave structure makes a team very resistant to the loss of a robot or a break in the communications. All the robots in a team will share the same distributed map and contribute to it as they move to unexplored areas. The correctness of the map is guaranteed through and dependent upon the continuous corrections, provided by SLAM, of the robots' positions.

The map generating algorithm used in this project will not rely on a single type of sensor and allows adding and removal of sensors as new sensors become available and old ones become obsolete. For all sensors used there is a corresponding sensor model which describes how the data from that particular sensor will contribute to the map [3]. The probabilistic nature of the sensor model makes the map generating algorithm more robust to sensor misreading.

The system of autonomously exploring robots is resilient to communication blackouts. If robots move through an area with bad radio reception, or if radio communication is altogether broken, individual robots continue exploring and gathering data. When communications are re-established, the gathered information is shared with the rest of the robots in the team.

6. Sensors for Navigation and Localization

All the sensors are connected to the central computer through a high speed CAN bus that runs through the robot. The use of sensor models allows the newly developed AI to interpret data from any sensor. From this year we are using the Player server as an interface to the hardware. By using the Player server our structure between the different sub parts of the system has improved. With the clear division of the sub parts it is made possible to implement them independently of each other.

Ultrasonic Sensors

The robots have 10 ultrasonic sensors which purpose is to supply information about distance to objects. The placement of the sensors is such that all directions around the robot are observed. The information is necessary for map generation and navigation of the robots and also for their ability to avoid collision.

The ultrasonic sensors that are used on the robots are the SRF04 Ultrasonic Ranger (see Fig 6.1.). The sensor card consists of transmitters and receivers. The transmitters emit a burst of 40 kHz sound when the control card issues a signal and the receivers listen for an echo. The approximate range of the ultrasonic ranger is from 3 cm to 3 m. The ultra sonic sensors are improved so that they are more stable and give more reliable data compared to last year.

Fig 6.1. SRF04 Ultrasonic Ranger.

Infrared Sensors

The infrared sensors complement the ultrasonic system. The characteristics in the environment sensed by the infrared and the ultra sonic sensors are very different. The analogue infrared sensors used operate on distances from 10 cm up to 80 cm and their accurate readings enable collision detection and avoidance at close distances. The infrared sensors are placed in a way such that all directions around the robot are observed. Using the orthogonal properties of the infrared and the ultrasonic sensors we are able to distinguish between transparent (e.g. plexiglass) and opaque obstacles.

Camera

Two cameras are mounted on a movable platform on top of the robot. Stereo algorithms are used to extract information from the camera images and this information is treated as an important sensor in the sensor model.

Motor Positioning Calculation

The electrical motors are controlled by a dedicated card, which also provides sensor data to the navigation system. To handle the communication with the dedicated card, the Player position interface is used.

A pair of AVR microcontrollers is mounted on the circuit board controller for the motors to take care of the pulses from the encoders and to make positioning calculations. The microcontrollers communicate through an i2c bus. The encoder information is being used to instantly adjust the motor speed and make a first estimate of the position of the robot. The solution with motor controller and encoder feedback on the same circuit board gives the possibility to drive the motors very exactly.

From the engine controller, CAN messages containing position information is being sent continuously. The pace of position information packets being sent is controlled by how the robot is moving. The position packets can also be sent on demand.

Position Recovery

The map generation and position adjusting algorithm described under “Map Generation/Printing” guarantees that robots positions in the map will not differ substantially from the true positions.

When two robots start at different and initially unknown positions in an area, they may have radio communication but no knowledge of their relative positions. This will also be the situation if a robot in a team is reset or moved to a new location. In this case, the robots maintain individual maps. If in this case a robot encounters another robot during its exploration, it will initiate an algorithm to fuse the maps built when these robots operated individually of each other. The map fusion is a completely automated process and no operator assistance is needed.

7. Sensors for Victim Identification

Pyro-electric Sensors

Pyro-electric sensors are used to provide a way for the robots to identify and localize victims. A pair of sensors is used in stereo to provide accurate information of the position of a victim.

The pyro-sensor is a passive infrared sensor responsive to the wavelength of IR light emitted by warmth (e.g. a live human body). Since the sensor reacts to differences in temperature between two areas, light from these two areas is focused upon the receptive areas of the sensor through a thin sliver of a fresnel lens, and a pair of sensors sweep their surroundings mounted on a stepper motor.

The stepper motor is controlled by an AVR microcontroller, which also interprets the analog signal from the sensors. The position of a victim is found by triangulation. The AVR then in turn communicates the positions of found victims to the robot’s central CPU via the CAN bus.

Microphone

A microphone is used to aid the localization of victims. This together with a carbon dioxide meter makes it possible to locate hidden or entombed living humans.

Audio is continuously sampled through a single directional microphone facing forward. The audio stream is then processed for features distinguishing human voices and tapping sounds from background noises.

Camera

Web cameras connected to the robot central computer are used. The cameras are amongst other things used to document sites where victims are discovered to further help the rescue operation. These cameras are capable of identifying shapes of human bodies. We use several LED lamps to be able to take pictures in dark environments to aid the SLAM system and make remote operation possible.

Carbon Dioxide Sensor

The robots are from this year equipped with a CO_2 sensor [5] which allows the robot to both make a more certain detection of visible victims but also to detect entombed victims. The sensor contains of a component board and a probe attached to it with a wire. As the CO_2 passes through the chamber in the probe, the concentration of the gas is measured. The sensor can detect concentrations from 0 up to 10000 ppm. Our sensor will be calibrated to detect CO_2 concentrations from 0 ppm to 2000 ppm.

8. Robot Locomotion

As shown in figure 8.1 the robots are constructed with two front driving wheels and one rear wheel. The rear wheel follows the robot like the wheels on a shopping cart or a wheelchair. The positive effects of using just one rear wheel compared to two rear wheels is that the robot size and turnaround radius decreases.

Two motors are used, one for each driving wheel. The robot turns by rotating the driving wheels at different velocities.

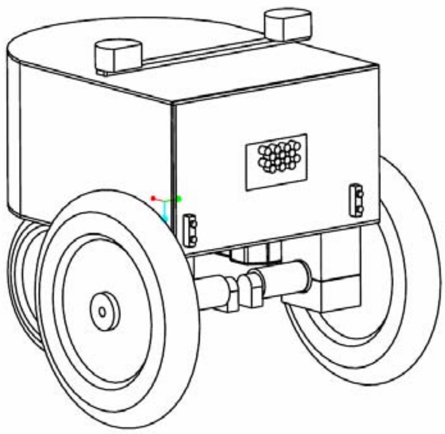

Fig. 8.1. The basic design of the robot.

9. Other Mechanisms

Motors
The motor packs consist of a motor, model RE40 from Maxon. The motor is a 150 Watt, 48 Volt DC motor. A planetary gear head is mounted on the motor. The motor is also equipped with a Maxon encoder which gives an output of 500 pulses per turn on the motor. The engines are strong enough to drive a robot weighing 20kg over obstacles up to 75mm high without problems.

The engines are controlled by a single circuit board controller. The board is equipped with a CAN interface to communicate with other nodes on the robot. A microcontroller is being used to handle the CAN communication and to control the motors.

Computer

The robots central computer is an embedded PC, based on a PC/104 CPU Board.

10. Team Training for Operation (Human Factors)

Since the system is fully autonomous minimal training is required for use. To operate the system a human/robot graphical interface is used where all robots can be controlled simultaneously. The interface is easy to use, but advanced functions for accessing robot subsystems for debugging purposes demands good knowledge of the system.

The operators don't need any extra training in addition to the normal testing before the competition. Tests however include setup time and how to handle failure on the robot subsystems that might force an operator to take control of the team. We select the person to operate the system in case of failure of the autonomous system by timed test runs.

11. Possibility for Practical Application to Real Disaster Site

Best chances for success are in buildings where ground obstacles are small. The robots don't have the need for external light sources because of the lamps they carry on board.

12. System Cost

The cost per robot is approximately 4000 €. This does not include the operator station, which consists of a standard PC with an IEEE802.11a WLAN card and a printer. For further information see appendix A.

References

1. Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In Proc. of the Intl. Conf. on Advanced Robotics (ICAR), pages 317-323, Coimbra, Portugal, July 2003.

2. Yamauchi, B., "A Frontier Based Approach for Autonomous Exploration," submitted to IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, July 10-11, 1997.

3 M.C. Martin and H. Moravec, Robot Evidence Grids, tech. report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University, March, 1996.

4. S. Thrun. Robotic mapping: A survey. In Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002.

5. http://www.vaisala.com/DynaGen_Attachments/Att18157/18157.pdf

Appendix A

Number of items	Module	tot price (EUR)
	WLAN card	
1	Orinoco Combocard Gold 802.11a/b Cardbus	**70,52**
	Webcam/mic	
2	Philips ToUcam Pro PCVC 740k	**130,64**
	Ultrasonic	
10	Devantech ultrasonic module	358
1	Control card	150
	Tot Ultrasonic	**508**
	Carbon dioxide	
1	Vaisala CARBOCAP Carbon Dioxide Module Series GMM220	**270,66**
	IR	
8	SHARP IR sensor	**38,94**
	Wheels	
2	Front wheel	31,35
1	Rear wheel	30,11
	Tot Wheels	**61,46**
	Motors (Stork)	
2	DC-motor	264,00
2	Planetary gear	234,30
2	Puls sensor	90,20
2	Assembly set	7,92
	Tot Motors	**596,42**
	Motor controller	
1	Parts	275,00
1	PCB	96,25
	Tot Motor controller	**371,25**
	The frame	
1	Work + material	**825,00**
	Batteries	
2	Batteries	126,50
1	Charger	24,75
2	Holds	5,50
	Tot Batteries	**156,75**
	Lamp	
4	LED	**20**

	Pyro-electric sensor	
2	Operational amplifier	1,28
2	Rail-to-rail op	1,70
2	Pyroelectric IR-sensor (Nippon Ceramic)	10,98
2	Fresnel lens (Nippon Ceramic)	10,03
2	Stepper motor driver (Allegro)	9,97
1	Stepper motor	56,32
	Tot Pyro	**90,28**
	RFC-CAN cards	
5	Cards	32,18
5	AVR Microcontroller	61,05
5	CAN controller	6,60
5	CAN transceiver	11,02
5	Reset circuit	8,18
6	Optocoupler	29,57
	Tot FRC-CAN cards	**228**
	CPU	
1	32MB SODIMM memory expansion	19,72
1	Hectronic H6015 central computer	221,54
1	CompactFlash 128MB	24,85
1	Hectronic H7006 CAN-card PC/104	38,50
1	Cables	33,00
1	Assembly – testing	16,50
	Tot CPU	**354,11**
	TOTAL SYSTEM COST	**3722**

RoboCup2005
Rescue Robot League Competition
Osaka, Japan
July 13 – 19, 2005
www.robocup2005.org

RoboCupRescue - Robot League Team
Team Savior Sphere, RFC Uppsala (Sweden)

Linus Granborg, Fredrik Österlind, Rickard Alfredson, Magnus Halvarsson, Serdar Ulusoy, Marcus Rosengren, Martin Spännar, Johan Snellman, Henrik Nilsson, David Shore, Per-Albin Jansson, Alexander Åhman

Uppsala University
Department of Information Technology
Box 337, SE-751 05 Uppsala, Sweden
jakob.carlstrom@it.uu.se

http://www.robocup.it.uu.se/itp2005/ratatosk/

Abstract. Savior Sphere is a semi-autonomous robot system based on an innovative spherical design. During one semester, twelve students have developed a wireless robot system for solving rescue tasks. The robot operates in disaster areas as a reconnaissance unit to explore an unknown and unsafe environment. Thanks to its spherical polycarbonated shell, the robot is robust to impact and does not get tangled in loose objects. An operator views a live video feed from the robot. At the same time the operator is presented with an automatically generated map of the surrounding area with indications of possible victims. This is accomplished by automatic fusion of data from dead reckoning, infrared, ultrasonic, CO_2 and pyro-electric sensors. The locomotion system uses a pendulum which changes the center of gravity. To make the movement smooth, an automatic control system and predefined advanced maneuvers developed by reinforcement learning are used.

Introduction

Savior Sphere provides a complete semi-autonomous system for rescue missions in unknown environments. With camera feed, map generation and victim health information, the robot helps rescue teams carry out their work in an efficient and safe way. The innovative spherical design enables it to access areas that are impossible for other robots. It can easily search muddy, watery or snowy areas. The robot shell is very durable to impacts which make it a perfect rescue robot. To learn advanced movement patterns, the robot has been trained using reinforcement learning. This way, the robot teaches itself how to get the best use of its capabilities. To speed up the training phase, a mechanical simulator has been developed and used. The operator will at all time be presented with a self-correcting map of the surrounding areas using SLAM techniques. The map can also be manually edited during missions for minor adjustments or to add waypoints. The user interface is designed to give the operator maximum control of the robot by presenting relevant information in a user-friendly way. All hardware, except the PC/104 computer, has been developed entirely by the team, such as, motor controller cards and CAN-controller cards.

Since year 2000 RFC Uppsala has been participating in RoboCup. In 2004 RFC Uppsala developed a new robot system, reusing experiences from previous years' projects, to participate in RoboCup rescue.

Fig. 1. Picture of the robot without motors, batteries, hardware and sensors.

As a complement to this paper there are two movies on the project webpage[1]. One movie shows the sphere prototype moving in snow. The second movie is a computer animation showing what the robot will look like and how it will act.

1. Team Members and Their Contributions

The Savior Sphere project team consists of twelve students from the Master of Science program in Information Technology and Electrical Engineering program at Uppsala University. The project is a fulltime course covering an entire semester. The team has several resource persons at Uppsala University who were in close collaboration throughout the project.

---

[1] See webpage http://www.robocup.it.uu.se/itp2005/ratatosk/

• Team Leader and advisor:	Jakob Carlström
• Advisor	Mikael Carlsson
• Advisor	Per Halvarsson
• Advisor	Mattias Wiggberg
• Project leader	Linus Granborg
• Technical Coordinator	Fredrik Österlind
• Hardware	Alexander Åhman, Serdar Ulusoy
• Communication	Henrik Nilsson, Marcus Rosengren
• Human-Computer- and Robot-human Interaction	Martin Spännar, David Shore
• Mapping	Fredrik Österlind, Rickard Alfredson
• Software	Magnus Halvarsson, Per-Albin Jansson
• Reinforcement learning	Johan Snellman
• Simulation	Johan Snellman, Henrik Nilsson

2. Operator Station Set-up and Break-Down (10 minutes)

A single person can easily manage to deploy the Savior Sphere. The robot weighs only about 8-10 kg and is easily carried into action in a bag. It is transported in one piece and needs no further assembly on site. Because of its robust construction in hard and flexible polycarbonate the robot can be dropped or even thrown into an area which is dangerous or inaccessible to humans.

Only a laptop computer and a joystick are needed to operate the Savior Sphere. Carried in a case along with the robot in a bag, the whole system weighs about 12-14 kg. When in place, the operator controls everything from a graphical user interface. No operational tasks are handled outside the computer.

3. Communications

The robot uses IEEE's 802.11a standard for wireless communication with the operator station. We use the TCP Westwood technique whose adaptive congestion control mechanism ensures fairness and throughput in the system. Compared to the Reno approach, which is used in most modern computers, our solution is significantly improved. The communication is set up in infrastructure mode, using an access point between the operator station and the robot. Communication with the robot is accomplished by utilizing UDP for the video stream and TCP Westwood for robot control messages.

Rescue Robot League		
TEAM SAVIOR SPHERE (RFC Uppsala SWEDEN)		
Frequency	**Channel/Band**	**Power (mW)**
5.0 GHz - 802.11a	Primary: 48/Middle Secondary: 36/Low 40/Low, 44/Low, 52/Middle, 56/Middle, 60/Middle, 64/Middle	Low: 50 mW Middle: 250 mW

The CAN-bus (Controller Area Network) is the most important bus on the robot, as all sensors and actuators are connected to it. It is a robust high-speed serial bus (up to 1 Mbit/s), often used as a communication network in cars. The CAN-bus was chosen because it has predictable behavior and response times. Another advantage is that it is a true multi-master bus, enabling many nodes to efficiently share the bus between them.

4. Control Method and Human-Robot Interface

A single operator controls the robot via the Human-Robot Interface. This is a Graphical User Interface (GUI) running on a laptop computer connected to a standard joystick. The GUI is used through all three stages of a mission: Setup, Actual Mission and Debriefing. Initialization, testing and status of each individual subsystem are managed.

During the actual mission, video and all sensor data gathered by the robot is presented to the operator. The operator can interact with the robot to confirm, correct or discard information. A map is automatically generated but editable by the operator. Potential victims are identified, localized and pointed out to the operator.

In the debriefing of a mission the GUI provides the operator with a summary. All mission data is combined to support an evaluation and a final mission report. To help guide the operator, the 2D map of the environment will also be mapped onto the video by using advanced graphical 3D transformations. This enables the operator to detect whether the map is correct simply by looking at the video. He may also correct the map by selecting areas in the 3D projection, and erase or move them to get a perfect map. The robot can be guided directly by using a joystick as well as by using more advanced, semi-automatic movement commands. These more advanced commands may be to turn the robot a certain numbers of degrees or move forward a specified length.

5. Map generation

At the same time as an operator guides the robot through an unknown terrain, the system will automatically generate and display a concurrent map of the surrounding area. To get a perfect result, the map is created via numerous different inputs. Infrared and ultra sonic sensors will cooperate and dead reckoning of the robot will be compared to sensor inputs. The operator will also at any time be able to correct the presented map. But the involvement of the operator will be at a minimum because the map corrects itself when detecting flaws in previous mapped environments.

To solve the concurrent mapping and localization problem, also referred as SLAM problem, Savior Sphere uses a hybrid solution of an incremental maximum likelihood estimator and a posterior pose estimator to combine the best of both worlds. This is chosen because it generates pseudo real time maps and also supports cyclic environments. The solution generate maps in real time by finding the most likely continuation of the previously maps under the most recent sensor measurements. However, when closing a loop, backward-correction is used to correct pose estimates backward in time and resolve inconsistencies. [1]

6. Sensors for Navigation and Localization

Infrared sensors are placed on specific points on both sides of the robot to detect objects and obstacles at short distance to ensure collision avoidance. The data from infrared sensors can be used by the motors to alter the robot’s course and is also used to create an accurate map of the robots immediate surroundings. These sensors are connected to an AVR microcontroller, which puts the sensor data on the CAN-bus.

Ultrasonic sensors are used to detect obstacles beyond the range of infrared sensors. At closer range they assist the Infrared sensors used for positioning. Their primary use is to detect transparent solid objects undetectable by the Infrared sensors. The sensors are placed on each side of the robot and their data is acquired by the AVR microcontroller and forwarded to the operator via the CAN-bus.

The electrical motors are controlled by a dedicated card, which also provides sensor data for navigation and map building. Together with accelerometers or inclinometers, positioned at specific points inside the sphere enough sensor data is acquired to obtain the movement of the sphere.

In order to obtain the most accurate information of the surrounding area, data from the infrared sensors, ultrasonic sensors and video are analyzed and compared with the data from the engines and accelerometers or inclinometers.

7. Sensors for Victim Identification

Two pyro-electric sensors, one on each side, are mounted on the robot and used to detect heat and localize victims. The pyro-electric sensor is a passive infrared sensor responsive to the wavelength of infrared light emitted by a warm human body. A thin silver fresnel lens focuses the incoming light from the different heat sources upon the receptive areas of the sensor. The signals from the sensors are amplified and captured by an AVR microcontroller which in turn sends it out on the CAN-bus.

Four video cameras are mounted on the robot: two on each side, one pair in the direction of the movement and one in the opposite. The operator only uses the pair of video cameras in the direction of the movement. A video multiplexer is used to select which pair of cameras to be used. These two videos streams are merged into one frame for the operator to view. The video cameras are equipped with ultra wide-angle lenses together enabling an almost 360 degree view around the robot. The video cameras (from OmniVision) deliver analogue signal out (RGB or composite). The MPEG4-card compresses the two analogue video signals from the active cameras and passes the data to the computer for transmission over the network to the operator.

All camera parameters and multiplexers are controllable from a camera controller connected to the CAN-bus. This enables the operator (or the computer) to adjust the video settings on-the-fly.

Video camera:
A total of four cameras are mounted on the robot

- Analogue, 1/3 inch color video chip
- Mono audio out
- RGB or YUV out
- Automatic exposure, gain and white balance
- All parameters controllable via I^2C-bus
- AVR camera controller connects local I^2C-bus to the CAN-bus

The video cameras also provide mono audio together with the video signal. The microphones are used to localize sound from hidden or visible victims and to analyze their health status. There is also a small loudspeaker to allow two-way communications with located victims.

Two CO_2 sensors are mounted on the robot, one on each side, to detect breathing victims. These sensors are sensitive to isobutane (i-C_4H_{10}), alcohol, sulfurdioxide SO_2 apart from carbon dioxide $CO_{2.}$ The readings are forwarded through the AVR microcontroller and the CAN-bus to the operator.

8. Robot Locomotion

The locomotion system is fully located within the sphere. The robot moves, accelerates, changes directions, and stops by means of a movable internal pendulum. When the pendulum moves inside the sphere, the robot's center of mass changes and causes it to roll. The direction of the pendulum is set by two motors working in two different planes. The robot can reach speeds of 20 miles per hour, go up inclines of at least 17 degrees and overcome obstacles with a height of 10 cm.

The closer the centre of mass is to the shell of the robot, the greater the potential acceleration and stability. For this reason the batteries and motors are placed on the pendulum near the shell.

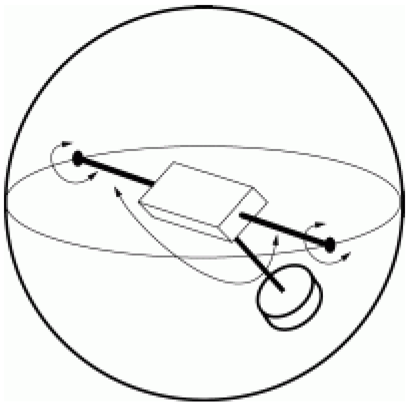

Fig. 2. A drawing of how the pendulum moves.

9. Other Mechanisms

Reinforcement learning
Certain movement patterns become difficult with a spherical robot. To solve this, reinforcement learning is used to allow the robot to find out for itself how to do complicated maneuvers. This allows the operator to use high-level commands to, for instance, make the robot turn around to get a 360 degrees view. [2]

Mechanical simulation
A mechanical simulator for robot movement has been developed. It simulates how the robot moves depending on motor inputs and current velocities. The simulator is used during reinforcement learning, when training the robot to perform various advanced movement patterns. This speeds up the learning process of the robot.

Automatic control
To stabilize the robot, and hence the cameras and sensors, an automatic control system for the engines has been designed. This also improves the accuracy of the steering and makes the robot easier to handle on uneven surfaces. [3]

Symmetrical design
Due to the symmetrical design, the robot does not have any given back or front side. Instead of turning around or trying to reverse, the operator simply switches to the opposite pair of cameras.

Hardware
The robot is equipped with a standard PC/104+ embedded computer responsible for collecting sensor data and making appropriate decisions.

An MPEG4 real-time encoder, 802.11a WLAN card and a CAN-bus controller are connected to the data bus of the computer. For hard disk emulation a standard compact flash card is used.

Main computer:

- Standard PC/104 form factor
- Linux operating system
- PentiumIII class processor, 500MHz
- 256 Mb SDRAM
- Solid state disk (CF-card)
- 802.11a WLAN-card (54 Mbps)
- 2 Video inputs for streaming video
- Real-time hardware MPEG4 encoder
- High speed CAN-controller

10. Team Training for Operation (Human Factors)

The GUI has been developed with an emphasis on making it easy to use by presenting data in an understandable way to a person with no prior knowledge of the system. No special skills or knowledge are therefore necessary to operate the system, only a familiarization of the controls and the visual aids of the user interface are needed.

A practice arena has been built to give the operator an understanding of the system and ultimately the ability to locate victims in a disaster area. In addition, the GUI can be connected to the USARSim simulator [4], where the robot has been modeled. This gives the operator the opportunity to use the system without access to the robot in a computer simulated area.

11. Possibility for Practical Application to Real Disaster Site

The nature of its design makes the Savior Sphere durable and endurable. It requires less maintenance than traditional robot constructions. The spherical polycarbonate shell is flexible and protects the sensitive robot parts inside, which gives it several advantages over traditional wheel- or track-based robots.

The Savior Sphere could be used for reconnaissance missions at a real disaster site. The robot is tough enough to handle rough conditions. The spherical shape allows moving through terrains such as gravel, mud or even snow. It is possible to make the robot totally water-proof, which would enable it to float.

12. System Cost

The total retail cost of the Savior Sphere is 5932 Euro. The actual cost was lower thanks to generous suppliers. The total cost does not include the operator station.

System cost		Price (Euro)
Robot mechanics		
	Motor (2)	778
	Mechanics	445
	Polycarbonate Cover	120
	Batteries (2)	27
	Construction	2450
Sum:		3820
Robot equipment		
	CO_2-sensor (2)	15
	Accelerometer (2)	64
	Pyro electrical sensor + lens (2)	26
	Ultrasonic sensors (2)	36
	IR-sensors (8)	199
	PC/104+	555
	MPEG card	500
	Video cameras (4)	222
	Motorcontroller card	330
	Additional electronics and circuit boards	165
Sum:		2112
Total cost:		5932

References

1. Thrun, Sebastian, “A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots”. *Int’l J. Robotics Research, 20(5):335-- 363, May 2001*.

2 Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge Massachusetts (1998)

3. Carstens, J.R.: Automatic Control Systems And Components. Prentice Hall, Englewood Cliffs New Jersey (1990)

4. USARSim website. http://usl.sis.pitt.edu/ulab/usarsim_download_page.htm